HW2

Andrew Rosen

October 7, 2013

1 Question 1

1.1 Part a

$$P = (C \boxplus \bar{K_1}) \oplus K_0$$

where $\bar{K_1}$ is the complement of K_1.

1.2 Part b

Let our plaintext messages be P_a, P_b and their corresponding ciphertexts be C_a, C_b. We know that

$$C_n = (P_n \oplus K_0) \boxplus K_1$$

We can define K_0 in terms of C_a, P_a, and K_1 by rearranging the variables.

$$(P_n \oplus K_0) \boxplus K_1 = C_n$$

$$P_a \oplus K_n = C_a \boxplus \bar{K_n}$$

$$K_0 = (C_n \boxplus \bar{K_1}) \oplus P_n$$

Which means you can't solve it without solving $\bar{K_1}$ or K_1.

$$(C_a \boxplus \bar{K_1}) \oplus P_a = (C_b \boxplus \bar{K_1}) \oplus P_b$$

$$(C_a \boxplus \bar{K_1}) \oplus P_a \oplus P_b = C_b \boxplus \bar{K_1}$$

$$\big((C_a \boxplus \bar{K_1}) \oplus P_a \oplus P_b\big) \boxplus C_b = \bar{K_1}$$

The solution is intractably self-referential. There is no easy solution for K_0 or K_1. Or at least non I've found after 5 hours of working on this problem.

2 Question 2

I will solve this problem for the general case. Let me denote the encryption of message m as $E(m)$, since the key is not relevent here. The defined encryption scheme being linear means

$$E(m_a) \oplus E(m_b) = E(m_a \oplus m_b)$$

This also implies that a message made of all zeroes will be encrypted to a message of all zeroes.

Now, let each input m_i and corresponding output $E(m_i)$ be l bits long. Choose l ciphertexts $E(m_i)$ such that

$$\begin{aligned} E(m_1) &= 1000\ldots \\ E(m_2) &= 0100\ldots \\ E(m_3) &= 0010\ldots \\ &\ldots \\ E(m_{l-1}) &= \ldots 0010 \\ E(m_l) &= \ldots 0001 \end{aligned}$$

I denote this set $\mathscr{E}$. If I know the corresponding plaintexts $\{m_1, m_2, \ldots m_l\}$, I can decipher any message using the linear property of E. Let $E(m_k)$ be an intercepted message. $E(m_k$ can be described by XORing a unique subset of $\mathscr{E}$. Now becuase of the linear property of E, I can retreive m_k by XORing the m's that correspond to the aforementioned unique subset.

For example, let

$$E(m_k) = E(m_1) \oplus E(m_{17}) \oplus E(m_{42})$$

We can retrieve m_k via

$$m_k = m_1 \oplus m_{17} \oplus m_{42}$$

3 Question 3

3.1 i

Since Bob already knows k and he knows the length of v, $(v||c)$ is effectively (v, c), yielding:

$$m = \text{RC4}(v||k) \oplus c$$

.

3.2 ii

We can detect reuse by trivially looking for two messages where the v's match.

4 Question 4

All RSA operations with a given key occur in the same mod space. For brevity, assume that every mathematical operation below has an unwritten mod n as a component.

This question can be solved via expansion. We have been given B_1 and B_2, as well as C_1, as well as key$\{e, n\}$. Let us solve for C_2 such that $RSAH(C_1, C_2) = RSAH(B_1, B_2)$.

$$RSAH(C_1, C_2) = RSA(RSA(B_1) \oplus B_2) = RSA(B_1^e \oplus B_2)$$

$$RSA(RSA(C_1) \oplus C_2) = RSA(B_1^e \oplus B_2)$$

$$(C_1^e \oplus C_2)^e = (B_1^e \oplus B_2)^e$$

$$C_1^e \oplus C_2 = B_1^e \oplus B_2$$

$$C_2 = C_1^e \oplus (B_1^e \oplus B_2)$$

Thus we can always choose a C_2 s.t. $RSAH(C_1, C_2) = RSAH(B_1, B_2)$, so $RSAH$ does not satisfy weak collision resistance $\blacksquare$.

5 Question 5

5.1 Message Integrity

In both cases Bob will see his computed $auth(x)$ won't match the sent $auth(x)$. That means x was modified.

5.2 Replay Attack

From Bob's perspective, Alice sent him 101 messages that just happen to be identical. He won't detect Oscar unless the flood of identical messages itself registers as unusual to him.

5.3 Sender Authentication with cheating 3rd party

Assume the question means that Bob has x and is trying to verify who the sent the message, Alice or Oscar (the other alternative would be solving $\exists x$, which is trivial because Bob just checks to see if x is there). A message signed by the senders private key can be decrypted with (and only with, given large enough keys), the corresponding public key. Bob can just check $auth(x)$ with Alice and Oscar's public keys to see who is telling the truth.

MACs are similarly checked, but with the shared key used to generate $auth(x)$ instead. So long as the keys Alice and Oscar are using aren't the same, it will be easy to verify who sent what.

5.4 Authentication with Bob cheating

If Alice sent x using a digital signature, then $auth(x)$ will be generated with her private key. If Alice can show that $auth(x)$ can only be verified via her public key, then Bob is lying.

Alice won't be able to prove this in the same way using MACs, however.

6 Question 6

Attached is a drawing, because inputs to inputs to inputs make more sense that way.